DVP-7410
1 Channel MPEG 1/2/4 Digital Video Decoder Card

Copyright
This documentation and the software included with this product are copyrighted 2004 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties which may result from its use.

Acknowledgments
IBM and PC are trademarks of International Business Machines Corporation. MS-DOS, Windows, Microsoft Visual C++ and Visual BASIC are trade-marks of Microsoft Corporation. Intel and Pentium are trademarks of Intel Corporation. Delphi and C++ Builder are trademarks of Inprise Corporation.

CE notification
The DVP-7410, developed by ADVANTECH CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information

On-line Technical Support
For technical support and service, please visit our support website at:
http://www.advantech.com/support

Part No. 20620000B0 1st Edition
Printed in Taiwan Jun. 2004
Rev. 0.1c

Contents

CHAPTER 1 GENERAL INFORMATION 1

1.1 HARDWARE REQUIREMENTS 2
1.2 SOFTWARE REQUIREMENTS 2
1.3 BLOCK DIAGRAM 3
Figure 1-1: Block Diagram 3
1.4 GPIO PIN DEFINITION 3
Table 1-1: GPIO Pin Definition 3
1.5 DIMENSIONS 4
Figure 1-2: Dimensions 4
1.6 HARDWARE INSTALLATION 4
1.7 SOFTWARE / DRIVER INSTALLATION 5

CHAPTER 2 FUNCTIONS LIBRARY 9

2.1 SUMMARY 10

CHAPTER 1

General Information

Chapter 1 General Information

The DVP-7410 is a single-chip audio/video decoder that supports SVCD, VCD1.x, XCD 2.0, CD-DA, and CD-ROM media formats as well as the ISMA (Internet Streaming Media Alliance) streaming format. Video decoding support includes MPEG-1, MPEG-2, MP@ML, and MPEG-4 advanced simple profile (without GMC and QPEL). Audio decoding support includes 16-bit MPEG-1 Layers1 and 2 (down mixed to stereo), and 16-bit linear PCM DVD support includes hardware CSS decryption, 16:9 and 4:3 aspect ratios, Letterboxing, 3:2 pull-down, multiple angles, and sub-picture. The programmable video output display controller includes higher-quality arbitrary scaling and an OSD frame buffer with 16-level alpha blending. Output resolutions up to 1920X1080, interlaced or progressive, are supported with a refresh rate up to 120 Hz. Brightness, contrast, and color controls are provided to optimize the displayed video.

1.1 Hardware Requirements

Intel Pentium III 800MHz or above (The CPU speed is depends on the video frame rate and resolution)
256MB RAM or above
CD-ROM
Hard disk with 128MB free space

1.2 Software Requirements

Microsoft Windows 98/ME/2000/XP with DirectX 8.1 or above

1.3 Block Diagram

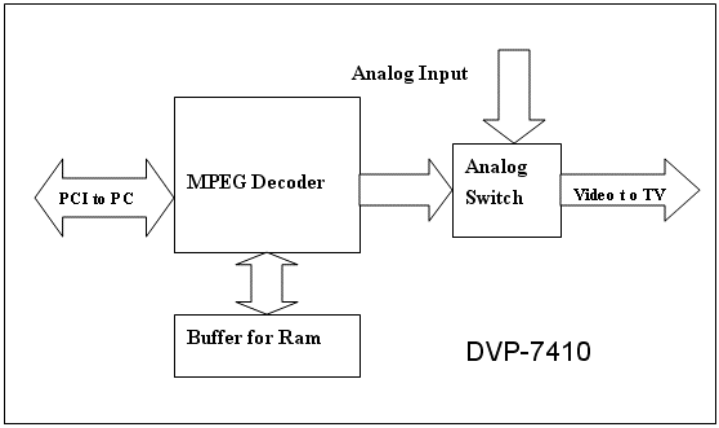


Figure 1-1: Block Diagram

1.4 GPIO Pin Definition

Table 1-1: GPIO Pin Definition

Pin no.	Description
0	I2C
1	I2C
2	Mute
3	Video Switch
4	VGA Switch
5	Card ID 0
6	Card ID 1
7	Card ID 2
8	Card ID 3
9	Protect Pull High
10	Protect Pull Low
11	Release
12	Release
13	Release
14	Release

1.5 Dimensions

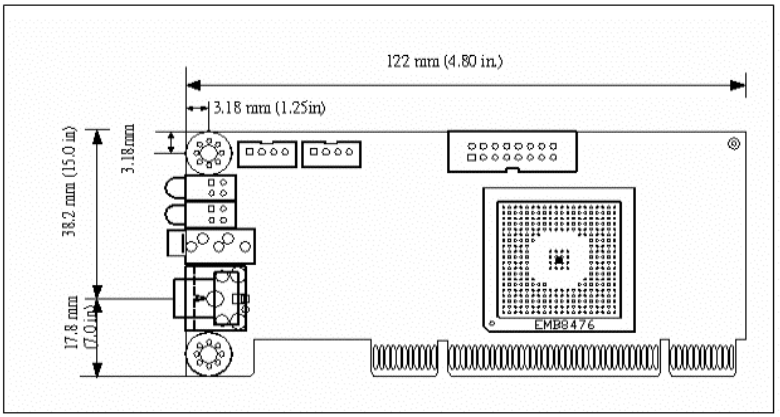


Figure 1-2: Dimensions

1.6 Hardware Installation

Turn off your computer and unplug the power cord.
Remove the cover of your computer.
Touch the metal part on the surface of your computer to neutralize the static electricity that might be on your body.
Place the DVP-7410 into chassis and using screws to fix it.
Connect appropriate accessories to the DVP-7410 decoder card.
Replace the cover of your computer chassis.
Plug in the power cord and turn on the computer.

Note: Keep the anti-static bag for future use. You might need the original bag to store the card if you have to remove the card from the PC or transport it elsewhere.

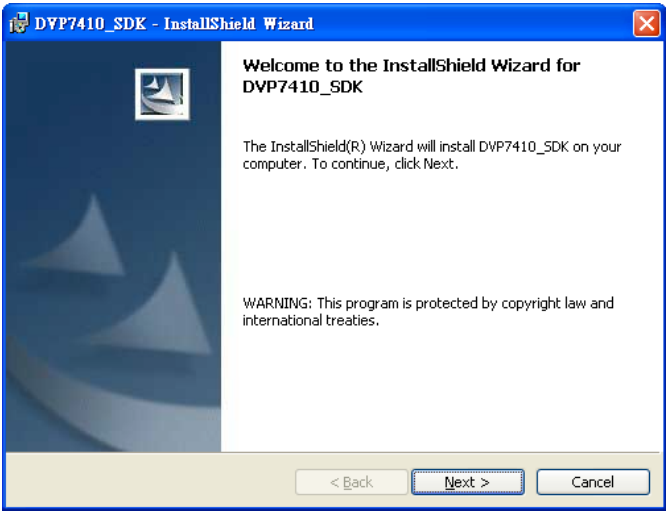
1.7 Software / Driver Installation

Before you begin
To facilitate the installation of the enhanced display device drivers and utility software, you should read the instructions in this chapter carefully before you attempt installation. The device drivers for the DVP-7410 board are located on the software installation CD. The auto-run function of the driver CD will guide and link you to the utilities and device drivers under Windows system. Before you begin, it is important to note that most display drivers need to have the relevant software application already installed in the system prior to installing the enhanced display drivers. In addition, many of the installation procedures assume that you are familiar with both the relevant software applications and operating system commands. Review the relevant operating system commands and the pertinent sections of your application software user's manual before performing the installation.

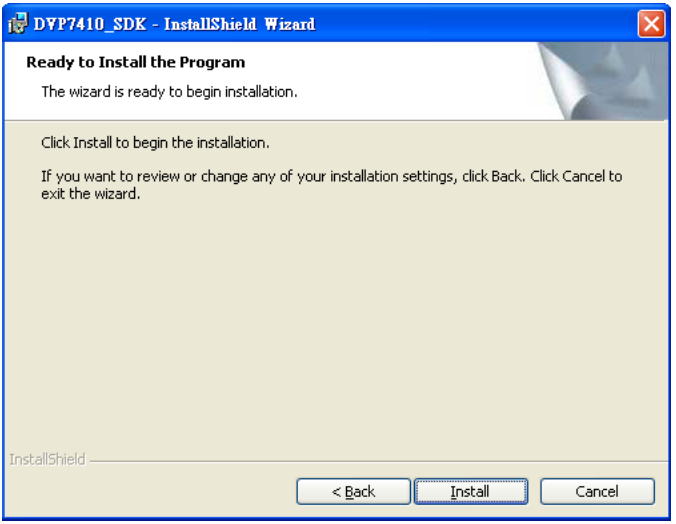
Installing

1. Insert the driver CD into your system's CD-ROM drive. In a few seconds, the software installation main menu appears. Move the mouse cursor over the "Manual" button under the "SETUP" heading, a message pops up telling you to start the installation.

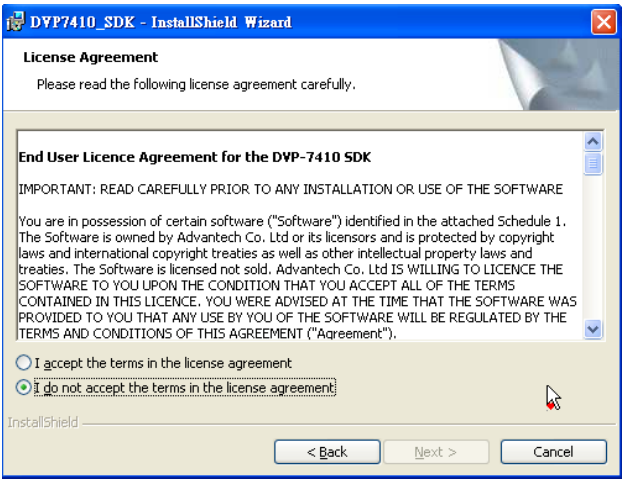
2. Click "Next" when you see the following message.



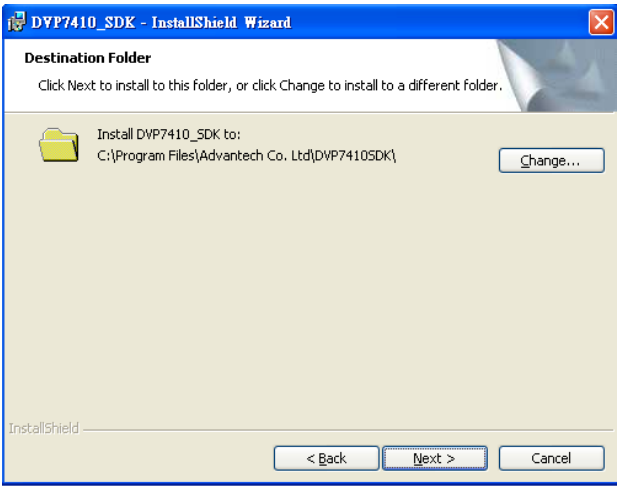
3. Click "Install" when you see the following Ready to Install the Program message.



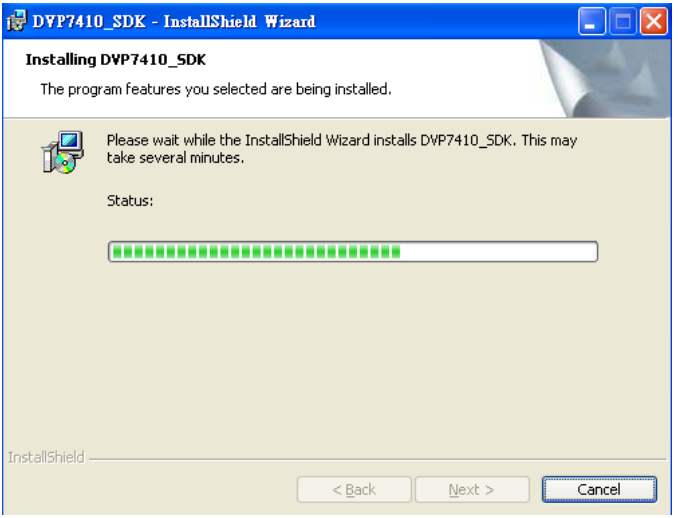
4. Please read the following license agreement and select "I accept the terms in the license agreement" to next status.



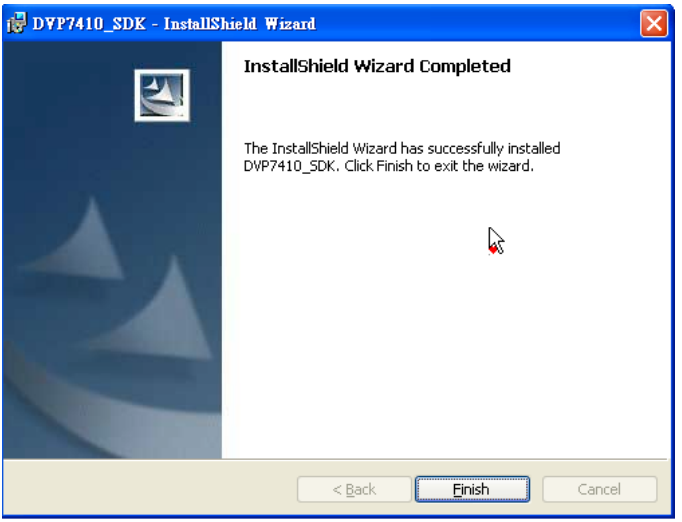
5. When the following message appear click “Next” or click “Change” to install to a different folder.



6. Installing DVP-7410 SDK.



7. When the following message appears, click "Finish" to complete the installation and restart Windows.



CHAPTER 2

Functions Library

Chapter 2 Functions Library

2.1 Summary

Initialize Part :

init_ADVSDK

Global Settings Part :

Adv_DVPAPI_SetSearchExtensions
Adv_DVPAPI_CreateFileList
Adv_DVPAPI_GetPlayListCount
Adv_DVPAPI_GetPlayList
Adv_DVPAPI_GetFileList
Adv_DVPAPI_LoadPlayList
Adv_DVPAPI_GetFileFromPlayList
Adv_DVPAPI_ResetPlayList
Adv_DVPAPI_AddFileToPlayList
Adv_DVPAPI_SavePlayList
Adv_DVPAPI_SetNotifyWindow
Adv_DVPAPI_GetFileListCount
Adv_DVPAPI_GetDisplayCategory
Video/Audio Control Part :

Adv_DVPAPI_SetPlaybackList
Adv_DVPAPI_InitController
Adv_DVPAPI_IsControllerInit
Adv_DVPAPI_UnInitController
Adv_DVPAPI_SetVideoInputReceiver
Adv_DVPAPI_GetFileType
Adv_DVPAPI_SetVideoPosition

Adv_DVPAPI_GetPlaybackMode
Adv_DVPAPI_Play
Adv_DVPAPI_Pause
Adv_DVPAPI_Stop
Adv_DVPAPI_Seek
Adv_DVPAPI_NextTrack

Adv_DVPAPI_PreviousTrack

Adv_DVPAPI_CopyFrame
Adv_DVPAPI_CaptureFrame

Adv_DVPAPI_GetVolumeLevel
Adv_DVPAPI_SetVolumeLevel
Adv_DVPAPI_GetPictureControlValue
Adv_DVPAPI_SetPictureControl

Adv_DVPAPI_GetVideoKeyColor
Adv_DVPAPI_SetVideoKeyColor

Adv_DVPAPI_GetLength
Adv_DVPAPI_GetVideoSourceSize
Adv_DVPAPI_GetVideoHandle
Adv_DVPAPI_GetVideoStandard
Adv_DVPAPI_GetFileInfo
Adv_DVPAPI_GetVideoPosition

Adv_DVPAPI_GetTVOutput

Adv_DVPAPI_GetTVScale
Adv_DVPAPI_SetTVScale

Adv_DVPAPI_GetTVStandard
Adv_DVPAPI_SetTVStandard
Adv_DVPAPI_GetDisplayMode
Adv_DVPAPI_SetDisplayMode

Adv_DVPAPI_GetHDTVMode
Adv_DVPAPI_SetHDTVMode
Adv_DVPAPI_GetHDTVModeSubD
Adv_DVPAPI_SetHDTVModeSubD
Adv_DVPAPI_GetDestinationInDisplay
Adv_DVPAPI_SetDestinationInDisplay

Adv_DVPAPI_GetAudioOutputMode
Adv_DVPAPI_SetAudioOutputMode
Adv_DVPAPI_GetAudioOutputType
Adv_DVPAPI_SetAudioOutputType

Adv_DVPAPI_GetNavigationMode
Adv_DVPAPI_SetNavigationMode
Adv_DVPAPI_GetRepeatMode
Adv_DVPAPI_SetRepeatMode

GPIO Part :

Adv_DVPAPI_SwitchTo7420
Adv_DVPAPI_MuteOn

Functions Reference

Type enumeration

DRIVE_ID
typedef enum
{
    DRIVE_CURRENT,
    DRIVE_UNDETERMINED,
    DRIVE_C, DRIVE_D, DRIVE_E, DRIVE_F, // Other drive letters
    DRIVE_G, DRIVE_H, DRIVE_I, DRIVE_J,
    DRIVE_K, DRIVE_L, DRIVE_M, DRIVE_N,
    DRIVE_O, DRIVE_P, DRIVE_Q, DRIVE_R,
    DRIVE_S, DRIVE_T, DRIVE_U, DRIVE_V,
    DRIVE_W, DRIVE_X, DRIVE_Y, DRIVE_Z

} DRIVE_ID, *LPDRIVE_ID;
Description
A typedef enumeration stores system drives information.

Parameters

DRIVE_CURRENT: Currently set drive (only used on input).
DRIVE_UNDETERMINED: No drive set.
DRIVE_C ~ DRIVE_Z: Other drive letters.
DISPLAY_CATEGORY

typedef enum
{
    RM_DISPLAY_HPLUS,

```
    RM_DISPLAY_NS2000,
    RM_DISPLAY_NS2000_NO,
    RM_DISPLAY_V2000,
    RM_DISPLAY_V2000_NVP,
    RM_DISPLAY_GALAXY2,
    RM_DISPLAY_XCARD,
    RM_DISPLAY_XCARD_NO,
    RM_DISPLAY_XCARD_NVP,
    RM_DISPLAY_XCARD_NOVP,
    RM_DISPLAY_SYMPHONY,
    RM_DISPLAY_SYMPHONY_NVP,
    RM_DISPLAY_NTSCPAL_CVBS
} DISPLAY_CATEGORY, *LPDISPLAY_CATEGORY;
```

Description
A typedef enumeration stores available display categories information.

Parameters

RM_DISPLAY_HPLUS : VGA, TV
RM_DISPLAY_NS2000 : VGA, TV+, HDTV Full Screen Sub-D
RM_DISPLAY_NS2000_NO: TV+, HDTV Sub-D, HDTV+
RM_DISPLAY_V2000 : TV+_VGA, HDTV, HDTV+, 480P
RM_DISPLAY_V2000_NVP : TV+, HDTV, HDTV+, 480P
RM_DISPLAY_GALAXY2 : TV_TVIA
RM_DISPLAY_XCARD : TV+_VGA, HDTV_FS, HDTV+, RGB_SCART
RM_DISPLAY_XCARD_NO : TV+_VGA, TV+, HDTV, HDTV+, RGB_SCART
RM_DISPLAY_XCARD_NVP : TV+, HDTV_FS, HDTV+, RGB_SCART
RM_DISPLAY_XCARD_NOVP : TV+, HDTV, HDTV+, RGB_SCART
RM_DISPLAY_SYMPHONY : TV+_VGA, HDTV+
RM_DISPLAY_SYMPHONY_NVP : TV+, HDTV+
RM_DISPLAY_NTSCPAL_CVBS : VGA NTSC or PAL

FILE_TYPE

```
typedef enum
{
```

```
    FILE_UNKNOWN_TYPE,
    FILE_MPEG_AUDIO,
    FILE_MPEG_VIDEO,
    FILE_MPEG_AUDIO_VIDEO,
    FILE_AC3,
    FILE_MP3,
    FILE_DVD_SOURCE,
    FILE_DIGITAL_AUDIO,
    FILE_VCD_STILL,
    FILE_MPEG4,
    FILE_DIVX,
    FILE_DTS_AUDIO
} FILE_TYPE, *LPFILE_TYPE;
```

Description

A typedef enumeration stores type of loaded file.

Parameters

FILE_UNKNOWN_TYPE : Unknown file type.
FILE_MPEG_AUDIO: MPEG audio only file.
FILE_MPEG_VIDEO: MPEG video only file
FILE_MPEG_AUDIO_VIDEO : MPEG video + audio (system) file
FILE_AC3 : AC3 source
FILE_MP3 : MP3 source
FILE_DVD_SOURCE : DVD source
FILE_DIGITAL_AUDIO : Audio CD source
FILE_VCD_STILL : VideoCD still picture
FILE_MPEG4 : MPEG4 source
FILE_DIVX : DIVX source
FILE_DTS_AUDIO : DTS source

PLAY_MODE

```
typedef enum
{
    MODE_UNKNOWN,
    MODE_NOT_INITIALIZED,
    MODE_INITIALIZED,
    MODE_STOPPED,
    MODE_PLAYING,
    MODE_PAUSED,
    MODE_RECORDING,
```

```
    MODE_PREVIEWING,
    MODE_TRICK_PLAY,
    MODE_STILL_MENU,
    MODE_WAIT_STOP,
    MODE_HW_SELECTED
} PLAY_MODE, *LPPLAY_MODE;
```

Description
A typedef enumeration stores play mode.

Parameters

MODE_UNKNOWN : Unknown mode
MODE_NOT_INITIALIZED : Not ready for playback
MODE_INITIALIZED : Ready for playback
MODE_STOPPED : Stopped
MODE_PLAYING : Playing
MODE_PAUSED : Paused
MODE_RECORDING : Recording
MODE_PREVIEWING : Previewing
MODE_TRICK_PLAY : FF/REW/Slow motion
MODE_STILL_MENU : Still menu mode,
MODE_WAIT_STOP : Uninitialized but expecting another disc (i.e. next VCD volume)
MODE_HW_SELECTED : Decoder has been selected

NAV_SEEK_MODE

```
typedef enum
{
    SEEK_FORWARD,
    SEEK_BACKWARD,
    SEEK_TO_START,
    SEEK_TO_POSITION
} NAV_SEEK_MODE, *LPNAV_SEEK_MODE;
```

Description
A typedef enumeration stores seek mode.

Parameters

SEEK_FORWARD : Seek forward
SEEK_BACKWARD : Seek backward

SEEK_TO_START : Seek to the beginning
SEEK_TO_POSITION : Seek to a given time position

PICTURE_CONTROL

```
typedef enum
{
    PCTL_BRIGHTNESS,
    PCTL_CONTRAST,
    PCTL_SATURATION,
    PCTL_HUE
} PICTURE_CONTROL, *LPPICTURE_CONTROL;
```

Description

A typedef enumeration stores picture control information.

Parameters

PCTL_BRIGHTNESS : Brightness
PCTL_CONTRAST : Contrast
PCTL_SATURATION : Saturation (Color)
PCTL_HUE : Hue -- not available on playback devices

VIDEO_STANDARD

```
typedef enum
{
    VIDEO_STD_UNKNOWN,
    VIDEO_STD_NTSC,
    VIDEO_STD_PAL,
    VIDEO_STD_SECAM
} VIDEO_STANDARD, *LPVIDEO_STANDARD;
```

Description

A typedef enumeration stores video standard information.

Parameters

VIDEO_STD_UNKNOWN : Unknown standard
VIDEO_STD_NTSC : NTSC
VIDEO_STD_PAL : PAL
VIDEO_STD_SECAM : SECAM

TV_OUT_MODE

```
typedef enum
{
    TV_OUT_OFF,
    TV_OUT_COMPOSITE,
    TV_OUT_COMPONENT_YUV,
    TV_OUT_COMPONENT_RGB,
    TV_OUT_RGB_SCART
} TV_OUT_MODE, *LPTV_OUT_MODE;
```

Description

A typedef enumeration stores TV out mode information.

Parameters

TV_OUT_OFF : TV display off

TV_OUT_COMPOSITE : TV & HDTV display through Composite/SVideo connector (default)

TV_OUT_COMPONENT_YUV : TV & HDTV display through Component YUV special connector

TV_OUT_COMPONENT_RGB : TV & HDTV display through Component RGB special connector (G contains Sync)

TV_OUT_RGB_SCART : TV & HDTV display through SCART connector (Separate CVBS)

TV_STANDARD

```
typedef enum
{
    TV_AUTO_STANDARD,
    TV_NTSC,
    TV_PAL,
    TV_PAL60,
} TV_STANDARD, *LPTV_STANDARD;
```

Description

A typedef enumeration stores TV standard information.

Parameters

TV_AUTO_STANDARD : Display to TV according to video source standard

TV_NTSC : Display NTSC to TV
TV_PAL : Display PAL to TV
TV_PAL60 : Display PAL 60Hz to TV
DISPLAY_MODE

```
typedef enum
{
    DISPLAY_VGA,
    DISPLAY_TV,
    DISPLAY_VGA_480P,
    DISPLAY_HDTV_SUBD,
    DISPLAY_HDTV
} DISPLAY_MODE, *LPDISPLAY_MODE;
```

Description
A typedef enumeration stores available display mode information.

Parameters

DISPLAY_VGA : Display on VGA using Analog Overlay
DISPLAY_TV : Display only on TV
DISPLAY_VGA_480P :
HDTV VGA output through Sub-D
connector 480P RGB only
DISPLAY_HDTV_SUBD :
HDTV VGA output through Sub-D
connector (always RGB)
DISPLAY_HDTV :
HDTV output (TV_OUT_MODE applies for YUV or RGB)
AUDIO_OUTPUT_MODE

```
typedef enum
{
    AUDIO_OUTPUT_STEREO,
    AUDIO_OUTPUT_LEFT_CHANNEL,
    AUDIO_OUTPUT_RIGHT_CHANNEL
} AUDIO_OUTPUT_MODE, *LPAUDIO_OUTPUT_MODE;
```

Description
A typedef enumeration stores audio output mode information.

Parameters

AUDIO_OUTPUT_STEREO : Stereo output
AUDIO_OUTPUT_LEFT_CHANNEL :
Left channel outputted through
both speakers or just left one
AUDIO_OUTPUT_RIGHT_CHANNEL :
Right channel outputted through
both speakers or just right one

NAVIGATION_MODE

```
typedef enum
{
    NAVIGATION_MODE_LIST,
    NAVIGATION_MODE_RANDOM,
    NAVIGATION_MODE_DVD,
    NAVIGATION_MODE_VCD,
    NAVIGATION_MODE_VCD2,
    NAVIGATION_MODE_SVCD,
    NAVIGATION_MODE_ENCODING
} NAVIGATION_MODE, *LPNAVIGATION_MODE;
```

Description
A typedef enumeration stores navigation mode information.

Parameters

NAVIGATION_MODE_LIST : Given list
NAVIGATION_MODE_RANDOM : Given list, randomly
NAVIGATION_MODE_DVD : Detected as DVD navigation
NAVIGATION_MODE_VCD : Detected as VideoCD
navigation
NAVIGATION_MODE_VCD2 : VideoCD 2.0 navigation
NAVIGATION_MODE_SVCD : SuperVcd navigation
NAVIGATION_MODE_ENCODING : Capture and/or Preview

REPEAT_MODE

```
typedef enum
{
```

REPEAT_OFF,
REPEAT_LIST,
REPEAT_CURRENT_FILE,
REPEAT_CURRENT_DVD_CHAPTER = 0x10,
REPEAT_CURRENT_DVD_TITLE
} REPEAT_MODE, *LPREPEAT_MODE;

Description

A typedef enumeration stores repeat mode information.

Parameters

REPEAT_OFF : No repeat mode set (default)
REPEAT_LIST : Repeat list in the default order
REPEAT_CURRENT_FILE : Repeat current file
REPEAT_CURRENT_DVD_CHAPTER = 0x10 : Repeat current DVD chapter
REPEAT_CURRENT_DVD_TITLE : Repeat current DVD title

Function Calls

init_ADVSDK

Syntax

int _stdcall init_ADVSDK(void **pp)

Description

This function creates SDK instance.

Parameters

pp

[out] A pointer to the SDK.

Return Value

S_OK : Success.
E_FAIL : Failure.

Adv_DVPAPI_SetNotifyWindow

Syntax

HRESULT Adv_DVPAPI_SetNotifyWindow(
HWND hwndNotify,
UINT NotificationID)

Description

Sets the user defined parameters needed to receive the navigation notification messages

Parameters

hwndNotify
[in] Handle to the window to receive notifications. NULL can be passed to halt already established notifications.
NotificationID
[in] Notification message ID. For this ID we recommend using a value in the Win32 WM_APP range (0x8000 through 0xBFFF), which are defined by the Win32 API as '*messages available for use by applications'*.

Return Value

E_FAIL : Failure.
E_INVALIDARG : *NotificationID* argument is invalid.
RME_INVALID_WINDOW : Given window is invalid.
RME_NO_NOTIFY_WINDOW : No window has been specified for status notification.
S_OK : Success.

Remarks

The notification messages the application can receive after using Adv_DVPAPI_*SetNotifyWindow,* come to the given window's callback procedure. The message's *WPARAM* parameter states a *NAVIGATION_STATUS* flag and *LPARAM* stores values pertinent to it.
The currently defined notification are:

WPARAM	*LPARAM*	*Description*
STATUS_MEDIA_FORMAT_CHANGE	New MEDIA_FORMAT	Notifies a MEDIA_FORMAT change
STATUS_DRIVE_STATUS_CHANGE	New DRIVE_STATUS	Notifies a DRIVE_STATUS change
STATUS_MODE_CHANGE	New PLAY_MODE	Notifies a PLAY_MODE change
STATUS_TIME_CHANGE	HIBYTE(HIWORD) = Hours, LOBYTE(HIWORD) = Minutes, HIBYTE(LOWORD) = Seconds	Playback timestamp changed

STATUS_FILE_INITIALIZED	HIWORD = FILE_TYPE LOWORD = VIDEO_STANDARD (if applicable)	New media has been readied for playback. Some of the media characteristics are returned
STATUS_PLAYBACK_ENDED	Reserved	The end of the current media or list of files has been reached.
STATUS_TRACK_CHANGE	LOWORD = New track number HIWORD = Undefined	A new track, from the playback list in use, has been loaded for playback
STATUS_FULLSCREEN_MESSAGE	Full Screen window Input message identifier	Notifies of an input message received by the full screen window
STATUS_HARDWARE_EJECT_REQUEST	Reserved	The user pressed the hardware eject button on the default drive during playback
STATUS_STREAM_ASPECT_RATIO	Input stream ratio: 0 = 4:3 1 = 16:9	Reported video aspect ratio
STATUS_TRACK_LOAD_FAILED	Path of file that failed to load	Reported when a given for playback file is missing or cannot be played.
STATUS_CAPTURE_AVAILABILITY	0 = No 1 = Yes	Reported during playback indicating if the frame capture functionality is available for the currently played file
STATUS_SYSTEM_RESUME	Reserved	System is resuming from suspended mode
STATUS_DVD_MENU_CHANGE	0 = Off 1 = On	A DVD menu is being displayed or is being released
STATUS_DVD_TITLE_CHAPTER_CHANGE	HIWORD = Title LOWORD = Chapter	A new DVD title and/or chapter is being played
STATUS_DVD_STILL_MODE_CHANGE	0 = Off 1 = On	A DVD still is being displayed or is being released
STATUS_DVD_ASPECT_RATIO_CHANGE	New DVD_ASPECT_RATIO	Notifies of a DVD volume's aspect ratio change
STATUS_DVD_REGION_ERROR	0 = Disc and decoder regions do not match 1 = Disc and DVD drive (RPC2) regions do not match 2 = No region has been set 4 = Region storage has been tampered with -1 = Unknown Error	DVD Region related errors
STATUS_DVD_LANGUAGE_CHANGE_AVAILABLE	0 = No 1 = Yes	Notifies if audio stream change is possible at the title / chapter being played
STATUS_VCD_TRACK_CHANGE	LOWORD = New track number, HIWORD = Navigation Flags	A new VCD track has been loaded for playback
STATUS_VCD_VOLUME_ENDED	LOWORD =Current volume numbe, HIWORD = Next volume (0 if none)	Reports that a VCD volume has finished playing and the next sequential volume (if any)
STATUS_CAPTURE_LENGTH_CHANGE	LPCAPTURE_LENGTH	Indicates the current length of a file being recorded

Adv_DVPAPI_IsDriveAvailable

Syntax
HRESULT Adv_DVPAPI_IsDriveAvailable(
DRIVE_ID Drive,
LPBOOL lpAvailable)

Description
Retrieves the availability of a given system drive letter

Parameters
Drive
[in] DRIVE_ID of the drive to query availability.
lpAvailable
[out] Boolean specifying if drive is available or not.

Return Value

The returned HRESULT can take one of the following values:

E_INVALIDARG : lpAvailable argument is invalid.
S_OK : Success.

Adv_DVPAPI_GetPlayListCount

Syntax
HRESULT Adv_DVPAPI_GetPlayListCount(
LPWORD lpwItems)

Description
Retrieves the number of items (paths) contained in the
Navigator's Play List.

Parameters
lpwItems
[out] Pointer the number of items in the play list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpwItems argument is invalid.

Remark
The maximum number of play list items is 500.

Adv_DVPAPI_GetPlayList

Syntax
HRESULT Adv_DVPAPI_GetPlayList(
LPBSTR List)

Description
Retrieves the list of files created by the CreateFileList method.

Parameters
List
[out] Pointer to the retrieved array of wide character string (BSTR) list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: List argument is invalid.

Remarks
For this operation to succeed, the caller must allocate the correct amount of list elements by using Adv_DVPAPI_GetFileListCount method.

Adv_DVPAPI_SetSearchExtensions

Syntax
HRESULT Adv_DVPAPI_SetSearchExtensions(
LPBSTR Extensions,
BYTE bItems)

Description
Sets the extensions to be used by the Adv_DVPAPI_CreateFileList method

Parameters
Extensions
[in] Array of extensions
bItems
[in] Number of extensions given.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: Extensions argument is invalid or
bItems > MAX_EXTENSIONS
Remarks
If no extensions are set by using this method, the navigator interface uses mpg as the only valid extension.

Adv_DVPAPI_CreateFileList

Syntax
HRESULT Adv_DVPAPI_CreateFileList(
DRIVE_ID Drive,
LPWORD lpwItems)

Description
Searches a given system drive for files matching the default, or given set of extensions, and creates a list with them.
Parameters
Drive
[in] DRIVE_ID of the system drive to be scanned for files
lpwItems
[out] Number of files found in given drive.
The maximum value returned is 500.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpwItems argument is invalid.
RME_NO_DRIVE_SELECTED : Given drive is undetermined.
RME_DRIVE_NOT_AVAILABLE : Given drive is not available.

Remarks
DRIVE_CURRENT can be used as first parameter to scan the drive currently in use by the Navigator interface.

Adv_DVPAPI_GetFileList

Syntax
HRESULT Adv_DVPAPI_GetFileList(
LPBSTR List)

Parameters
List
[out] Pointer to the retrieved array of wide character string (BSTR) list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: List argument is invalid.

Description
Retrieves the list of files created by the Adv_DVPAPI_CreateFileList method.

See Also

Adv_DVPAPI_LoadPlayList

Syntax
HRESULT Adv_DVPAPI_LoadPlayList(
BSTR lpPath)

Description
Populates the Navigator's play list with paths stored in a given file.

Parameters
lpPath
[in] Full path of the file containing the list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpPath argument is invalid.

Adv_DVPAPI_GetFileFromPlayList

Syntax
HRESULT Adv_DVPAPI_GetFileFromPlayList(
WORD wIndex,
BSTR lpPath)

Description
Retrieves the file path of a given play list index.

Parameters
wIndex
[in] Play List index of path to be retrieved.
lpPath
[out] Retrieved wide character string (BSTR) path.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: One or both of the arguments is(are) invalid.

Adv_DVPAPI_SavePlayList

Syntax
HRESULT Adv_DVPAPI_SavePlayList(
BSTR lpPath)

Description
Saves the Navigator's Play List to the given file.

Parameters
lpPath
[in] Full wide character string (BSTR) path of file to contain
the list.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpPath argument is invalid.

Remarks
This method does not verify if the given file exists prior to opening it for writing.

Adv_DVPAPI_ResetPlayList

Syntax
HRESULT Adv_DVPAPI_ResetPlayList(void)

Description
Empties the Navigator's play list.

Parameters
This method has no parameters.

Return Value
S_OK: Success.
E_FAIL: Failure.
See Also

Adv_DVPAPI_AddFileToPlayList

Syntax
HRESULT Adv_DVPAPI_AddFileToPlayList(
BSTR lpPath)

Description
Adds a playback file path to the end of the Play List kept by the Navigator.

Parameters
lpPath
[out] Complete wide character string (BSTR) path to file to be added to the bottom of the navigator's play list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpPath argument is invalid.
E_ACCESSDENIED:Play List already contains the maximum number of elements (500).

Adv_DVPAPI_GetFileListCount

Syntax
HRESULT Adv_DVPAPI_GetFileListCount(
LPWORD lpwItems)

Description
Retrieves the number of files contained in the list created by the CreateFileList method.

Parameters
lpwItems
[out] Number of files in the currently scanned list.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: pwItems argument is invalid.

Remarks
The maximum number of files that can be contained in the Navigator interface list is 500.

Adv_DVPAPI_GetDisplayCategory

Syntax
HRESULT Adv_DVPAPI_GetDisplayCategory(
LPDISPLAY_CATEGORY lpDisplayCategory)

Description
Provides the display capabilities of the installed DVP7410 components.

Parameters
lpDisplayCategory
[out] Returned category

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpDisplayCategory argument is invalid.

Adv_DVPAPI_SetPlaybackList

Syntax
HRESULT Adv_DVPAPI_SetPlaybackList(
LPBSTR lpList,
WORD wListItems)

Description
Stores and sets a given file list.

Parameters
lpList
[in] Pointer to an array of NULL terminated wide character
string (BSTR) file paths.
wListItems
[in] Number of file paths in the given list.
The maximum number of paths is 500.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpList argument is invalid or wListItems is out of range.
E_ACCESSDENIED : This interface is not initialized.

Remarks

Only MEDIA_FORMAT_UNKNOWN or MEDIA_FORMAT_DATA formats require lists to be given to the Navigator Control. The list must be given prior to attempting to initialize the Navigator Control.
To replace an already initialized list, the Navigator Control must be uninitialized -- see Adv_DVPAPI_UnInitController.
A playback list can be prepared with help of the Navigator interface method Adv_DVPAPI_CreateFileList

Adv_DVPAPI_InitController

Syntax
HRESULT Adv_DVPAPI_InitController(
HWND hwndVideoParent)

Description
Loads the media to be played and readies the Navigator for playback.

Parameters
hwndVideoParent
[in] Parent of the video playback window. If the media contains video and this parameter is NULL, the video window is created as a POPUP window. If media is known as having no video this parameter is ignored.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_FALSE: This interface is already initialized.
Remarks
This method loads the device drivers in charge of handling the given media and only needs to be called once after the interface has been instantiated.
Media formats such as MEDIA_FORMAT_UNKNOWN and MEDIA_FORMAT_DATA, require that a file list is passed to the Navigator Control prior to calling this method. For this, use the Adv_DVPAPI_SetPlaybackList method. Adv_DVPAPI_InitController will return E_FAIL if no playback list is given. If a playback list needs to be changed and the Navigator Control interface is already initialized, call Adv_DVPAPI_UnInitController and Adv_DVPAPI_SetPlaybackList (in this order) prior to calling Adv_DVPAPI_InitController again.

Adv_DVPAPI_IsControllerInit

Syntax

HRESULT Adv_DVPAPI_IsControllerInit(
LPBOOL lpInitialized)

Description
Returns a flag indicating if the Navigator control, interface is in initialized state.

Parameters
lpInitialized
[out] Pointer to the interface's initialized state.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpInitialized argument is invalid.

Remarks
Adv_DVPAPI_GetPlaybackMode can also be used to know if the interface has been initialized. Initialization would be indicated by returned playback modes other than MODE_UNKNOWN and MODE_NOT_INITIALIZED.

Adv_DVPAPI_UnInitController

Syntax
HRESULT Adv_DVPAPI_UnInitController(void)

Description
Aborts current navigation closing any devices in use.

Parameters
This function has no parameters
Return Value
E_FAIL: Failure.
S_OK: Success.
E_FALSE: This interface is already uninitialized.

Adv_DVPAPI_SetVideoInputReceiver

Syntax
HRESULT Adv_DVPAPI_SetVideoInputReceiver(
const HWND hwndReceiver)

Description

Sets a window to receive forwarded video window input messages.

Parameters
hwndReceiver
[in] Handle of the window to forward input messages received by the video window. This parameter cannot be NULL.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: hwndReceiver argument is invalid.

Remarks
This method will post input messages received by the video window to the caller provided window. The messages, which are posted with their corresponding WPARAM and LPARAM parameters, include:
WM_KEYDOWN
WM_KEYUP
WM_LBUTTONDBLCLK
WM_LBUTTONDOWN
WM_LBUTTONUP
WM_MBUTTONDBLCLK
WM_MBUTTONDOWN
WM_MBUTTONUP
WM_MOUSEACTIVATE
WM_MOUSEMOVE
WM_NCLBUTTONDBLCLK
WM_NCLBUTTONDOWN
WM_NCLBUTTONUP
WM_NCMBUTTONDBLCLK
WM_NCMBUTTONDOWN
WM_NCMBUTTONUP
WM_NCMOUSEMOVE
WM_NCRBUTTONDBLCLK
WM_NCRBUTTONDOWN
WM_NCRBUTTONUP
WM_RBUTTONDBLCLK
WM_RBUTTONDOWN
WM_RBUTTONUP
For more information about these messages, please refer to the Win32 documentation..

Adv_DVPAPI_GetFileType

Syntax
HRESULT Adv_DVPAPI_GetFileType(
LPFILE_TYPE lpfType)

Description
Retrieves video / audio and or format of the currently loaded file.

Parameters
lpfType
[out] Pointer to the returned file type.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpfType argument is invalid.
E_ACCESSDENIED : This interface is not initialized.
Adv_DVPAPI_SetVideoPosition

Syntax
HRESULT Adv_DVPAPI_SetVideoPosition(
LPRECT lprPosition)

Description
Sets the position of the video window relative to its parent's client area.

Parameters
lprPosition
[in] Pointer to the rectangular position to be taken
by the video frame within the parents client area.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lprPosition argument is invalid.
E_ACCESSDENIED : This interface is not initialized.

Adv_DVPAPI_GetPlaybackMode

Syntax

HRESULT Adv_DVPAPI_GetPlaybackMode(
LPPLAY_MODE lpMode)

Description
Retrieves the current playback mode (Playing, Paused, Stopped, etc.) .

Parameters
lpMode
[out] Pointer to the current playback mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG: lpMode argument is invalid.

Adv_DVPAPI_Play

Syntax
HRESULT Adv_DVPAPI_Play(void)

Description
Starts playback.

Parameters
This function has no parameters
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

Remarks
Use this method to resume playback from paused or stopped mode.

Adv_DVPAPI_Pause

Syntax
HRESULT Adv_DVPAPI_Pause(void)

Description
Pauses playback.

Parameters
This function has no parameters

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_NOTIMPL : Method is not supported.
S_FALSE : Media is not in play mode or already in paused mode.

Adv_DVPAPI_Stop

Syntax
HRESULT Adv_DVPAPI_Stop(void)

Description
Stops playback (like pause but in stopped mode).

Parameters
This function has no parameters

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
Remarks
Use Adv_DVPAPI_Play to restart playback from stopped mode.

Adv_DVPAPI_Seek

Syntax
HRESULT Adv_DVPAPI_Seek(
NAV_SEEK_MODE Mode,
UINT uSecs)

Description
Changes the playback position.

Parameters
Mode
[in] NAV_SEEK_MODE direction the seeking is to take.
uSecs
[in] Number of seconds, from current location, to seek to.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

Remarks
usecs is ignored if the requested seek mode is SEEK_TO_START.
When NAV_SEEK_MODE is set to SEEK_TO_POSITION, usecs specify the absolute position in seconds to seek to.

Adv_DVPAPI_NextTrack

Syntax
HRESULT Adv_DVPAPI_NextTrack(void)

Description
Loads the next file from media depending on a given playback list, or the next sequence on other media.

Parameters
This function has no parameters
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : No playback list has been provided for media that requires it, or the interface has not initialized on media not requiring a playback list.

Remarks
If the Navigation Mode is set to NAVIGATION_MODE_LIST, this method moves the playback list index forward without the interface having to be initialized. If prior to calling this method the playback list is at the last index, calling Adv_DVPAPI_NextTrack sets the index to the first item in the list.
If the Navigation Mode is set to NAVIGATION_MODE_RANDOM this method moves the playback list index to the next randomly generated index.
If the media was playing when this method was called the next file starts in play mode, otherwise in stopped mode.

Adv_DVPAPI_PreviousTrack

Syntax
HRESULT Adv_DVPAPI_PreviousTrack(void)

Description
Loads the previous file from media depending on a given playback list, or the previous sequence on other media.

Parameters
This function has no parameters

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : No playback list has been provided for media that requires it, or the interface has not initialized on media not requiring a playback list.
Remarks
If the Navigation Mode is set to NAVIGATION_MODE_LIST, this method moves the playback list index back without the interface having to be initialized. If prior to calling this method the playback list is at the first index, calling Adv_DVPAPI_PreviousTrack sets the index to the last item in the list.
If the Navigation Mode is set to NAVIGATION_MODE_RANDOM this method moves the playback list index to the next randomly generated index.
If the media was playing when this method was called the next file starts in play mode, otherwise in stopped mode.

Adv_DVPAPI_CopyFrame

Syntax
HRESULT Adv_DVPAPI_CopyFrame(void)

Description
Copies the contents of the video buffer to the Window's Clipboard in 24 bit bitmap format.

Parameters
This function has no parameters.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
Remarks
The return value for calling this method while being out of context i.e. playing audio, is E_FAIL

Adv_DVPAPI_CaptureFrame

Syntax
HRESULT Adv_DVPAPI_CaptureFrame(
BSTR lpPath)

Description
Saves the contents of the video buffer to a given file in 24bit bitmap format.

Parameters
lpPath
[in] Pointer to a NULL terminated wide character buffer containing the path of the file to be saved. The path length must not exceed MAX_PATH.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpPath argument is invalid or too long.

Remarks
The return value for calling this method while being out of context i.e. playing audio, is E_FAIL

Adv_DVPAPI_GetVolumeLevel

Syntax
HRESULT Adv_DVPAPI_GetVolumeLevel(

LPDWORD lpdwLevel)

Description
Retrieves the current audio volume level.

Parameters
lpdwLevel
[out] Pointer to the currently set volume level.
The volume level varies between 0 and 1000.
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpdwLevel argument is invalid.
Adv_DVPAPI_SetVolumeLevel

Syntax
HRESULT Adv_DVPAPI_SetVolumeLevel(
DWORD dwLevel)

Description
Sets the audio volume level

Parameters
dwLevel
[in] New volume level. The level varies from 0 to 1000.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : dwLevel argument is out of range.
E_NOTIMPL : No method is supported.

Adv_DVPAPI_GetPictureControlValue

Syntax
HRESULT Adv_DVPAPI_GetPictureControlValue(
PICTURE_CONTROL picCtl,
LPDWORD lpdwValue)

Description
Retrieves the current values for Brightness, Contrast, Saturation or Hue.

Parameters
picCtl
[in] Picture control to be modified.
lpdwValue
[out] Picture control value. The valid range for these controls is from 0 to 1000.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpdwValue argument is invalid.

Adv_DVPAPI_SetPictureControl

Syntax
HRESULT Adv_DVPAPI_SetPictureControl(
PICTURE_CONTROL PictureControl,
DWORD dwValue)

Description
Sets the current value for Brightness, Contrast, Saturation or Hue.

Parameters
Mode
[in] Picture control to be adjusted.
dwValue
[in] New value for the control.
Adjustable range varies from 0 to 1000.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : dwValue argument is out of range.
E_NOTIMPL : No support for chosen picture control.

Remarks
Currently, video playback devices do not support HUE.

Adv_DVPAPI_GetVideoKeyColor

Syntax
HRESULT Adv_DVPAPI_GetVideoKeyColor(
LPCOLORREF lpColor)

Description
Retrieves current video key color.

Parameters
Lpcref
[out] Pointer to the currently set video key color.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : Lpcref argument is invalid.

Adv_DVPAPI_SetVideoKeyColor

Syntax
HRESULT Adv_DVPAPI_SetVideoKeyColor(
COLORREF cref)

Description
Sets the color to be used by the Navigator as default for video display.

Parameters
Cref
[in] New video key color.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : Cref argument is invalid.
E_NOTIMPL : No method is supported.

Adv_DVPAPI_GetLength

Syntax
HRESULT Adv_DVPAPI_GetLength(
LPDWORD lpdwLength)

Description
Retrieves the length of the currently loaded media in milliseconds.

Parameters
lpdwLength
[out] Pointer to currently loaded file length in milliseconds.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpdwLength argument is invalid.

Adv_DVPAPI_GetVideoSourceSize

Syntax
HRESULT Adv_DVPAPI_GetVideoSourceSize(
LPPOINT lpSize)

Description
Retrieves the non-scaled size of the currently loaded video file.

Parameters
lpSize
[out] Pointer to the returned width and height of the currently loaded video file.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpSize argument is invalid.

Adv_DVPAPI_GetVideoHandle

Syntax
HRESULT Adv_DVPAPI_GetVideoHandle(
HWND *hwnd)

Description
Retrieves the handle of the current video window.

Parameters

hwnd
[out] Returned video handle.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : *hwnd argument is invalid.

Remarks
The video window is owned and maintained by the Navigator Control interface and therefore the caller must not attempt to destroy it.

Adv_DVPAPI_GetVideoStandard

Syntax
HRESULT Adv_DVPAPI_GetVideoStandard(
LPVIDEO_STANDARD lpVidStd)

Description
Retrieves the video standard of the currently loaded file based on its source size

Parameters
lpVidStd
[out] Pointer to the standard used to create the currently loaded file.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpVidStd argument is invalid.

Remarks
This method uses the video source size to determine the video standard, therefore SECAM files are returned as VIDEO_STD_PAL. Files of widths smaller than 120 pixels are returned as VIDEO_STD_UNKNOWN.

Adv_DVPAPI_GetFileInfo

Syntax

HRESULT Adv_DVPAPI_GetFileInfo(
BSTR lpPath,
LPWORD lpwFile_no,
LPWORD lpwFiles)

Description
Retrieves name and position (relative to the current list) of the currently loaded file.

Parameters
lpPath
[out] Pointer to a NULL terminated wide character string to contain the currently loaded file path. This parameter can be NULL if not required. The path length must not exceed MAX_PATH.
lpwFile_no
[out] Pointer to the returned 1 based index of the file being played. This parameter can be NULL if not required.
lpwFiles
[out] Pointer to the returned number of files in the current list. This parameter can be NULL if not required.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG : No arguments were requested.

Adv_DVPAPI_GetVideoPosition

Syntax
HRESULT Adv_DVPAPI_GetVideoPosition(
LPRECT lprPosition)

Description
Retrieves the current position of the video window relative to its parent's client area

Parameters
lprPosition
[out] Pointer to the rectangular region occupied by the video window relative to its parent's client area

Return Value
E_FAIL: Failure.

S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lprPosition argument is invalid.
Adv_DVPAPI_GetTVOutput

Syntax
HRESULT Adv_DVPAPI_GetTVOutput(
LPTV_OUT_MODE lpTVOut)

Description
Retrieves the currrent TV output mode.

Parameters
lpTVOut
[out] Pointer to the currently set TV output mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpTVOut argument is invalid.

Adv_DVPAPI_GetTVScale

Syntax
HRESULT Adv_DVPAPI_GetTVScale(
PBOOL lpbScaled)

Description
Retrieves whether or not the TV standard conversion is currently using scaling.

Parameters
lpbScaled
[out] TRUE if scales, FALSE otherwise

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpbScaled argument is invalid.

Adv_DVPAPI_SetTVScale

Syntax
HRESULT Adv_DVPAPI_SetTVScale(
BOOL bScaled)

Description
Sets whether or not the TV standard conversion is to use scaling. This API is used to prevent NTSC to PAL or PAL to NTSC field jitter artifacts due to scaling.

Parameters
bScaled
[in] TRUE to scale video, FALSE otherwise.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

Adv_DVPAPI_GetTVStandard

Syntax
HRESULT Adv_DVPAPI_GetTVStandard(
LPTV_STANDARD lpTVStandard)

Description
Retrieves the currently set TV output standard.

Parameters
lpTVStandard
[out] Set TV standard.
Adv_DVPAPI_SetTVStandard

Syntax
HRESULT Adv_DVPAPI_SetTVStandard(
TV_STANDARD tvStandard)

Description
Sets the default TV output mode.

Parameters
tvStandard
[in] Desired TV out standard

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

Adv_DVPAPI_GetDisplayMode

Syntax
HRESULT Adv_DVPAPI_GetDisplayMode(
LPDISPLAY_MODE lpMode)

Description
Retrieves the current mode of the display, being VGA or any of the supported TV modes.

Parameters
lpMode
[out] Pointer to the returned display mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpMode argument is invalid.

Adv_DVPAPI_SetDisplayMode

Syntax
HRESULT Adv_DVPAPI_SetDisplayMode(
DISPLAY_MODE Mode)

Description
Sets the display mode to be VGA or any of the supported TV modes.

Parameters
Mode
[in] New display mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_NOTIMPL: Method is not supported.

Remarks
TV display has to be already enabled in order to set it to a particular standard/scale mode. So for instance, to set the TV display to PAL from VGA only mode, first call Adv_DVPAPI_SetDisplayMode(DISPLAY_TV) and then Adv_DVPAPI_SetDisplayMode(DISPLAY_TV_PAL).

Adv_DVPAPI_GetHDTVMode

Syntax
HRESULT Adv_DVPAPI_GetHDTVMode(
PINT piHDTVModeIndex)

Description
Retrieves the set HDTV mode.

Parameters
piHDTVModeIndex
[out] HDTV mode index, were:

0 : 480p / 59.94Hz
1: 720p / 60Hz
2: 1080i / 60Hz

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : piHDTVModeIndex argument is invalid.

Adv_DVPAPI_SetHDTVMode

Syntax
HRESULT Adv_DVPAPI_SetHDTVMode(
INT iHDTVModeIndex)

Description
Selects the HDTV mode index based on a list of modes taken from Sigma registry (0..n). These modes include display resolution and monitor frequency.

Parameters
iHDTVModeIndex
[in] HDTV mode index

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
Remarks
Adv_DVPAPI_SetDisplayMode needs to be called using DISPLAY_HDTV for this mode to become effective.

Adv_DVPAPI_GetHDTVModeSubD

Syntax
HRESULT Adv_DVPAPI_GetHDTVModeSubD(
PINT piHDTVModeIndex)

Description
Retrieves the set HDTV SubD mode.

Parameters
piHDTVModeIndex
[out] HDTV SubD mode index

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : piHDTVModeIndex argument is invalid.

Adv_DVPAPI_SetHDTVModeSubD

Syntax
HRESULT Adv_DVPAPI_SetHDTVModeSubD(
INT iHDTVModeIndex)

Description
Selects the HDTV SubD (VGA connector) mode index based on a list of modes taken from Sigma registry (0..n). These modes include display resolution and monitor frequency.

Parameters
iHDTVModeIndex
[in] HDTV SubD mode
Return Value
E_FAIL: Failure.

S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

Remarks
Adv_DVPAPI_SetDisplayMode needs to be called using
DISPLAY_HDTV_SUBD for this mode to become effective

Adv_DVPAPI_GetDestinationInDisplay

Syntax
HRESULT Adv_DVPAPI_GetDestinationInDisplay(
LONG *pLeft,
LONG *pTop,
LONG *pWidth,
LONG *pHeight)

Description
Retrieves video's destination rectangle within the display device.

Parameters
pLeft
[out] Video's left coordinate
pTop
[out] Video's right coordinate
pWidth
[out] Video's width
pHeight
[out] Video's height
Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : One or more of the arguments are invalid.
E_NOTIMPL : Method is not supported.
E_UNEXPECTED : Unexpected error.

Remarks
All returned values are in pixels

Adv_DVPAPI_SetDestinationInDisplay

Syntax
HRESULT Adv_DVPAPI_SetDestinationInDisplay(

LONG Left,
LONG Top,
LONG Width,
LONG Height)

Description
Sets video's destination rectangle within the display device. Mostly used to control the aspect ratio on TV, HDTV or VGA devices.

Parameters
Left
[in] Desired video's left coordinate
Top
[in] Desired video's top coordinate
Width
[in] Desired video's width
Height
[in] Desired video's height

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_NOTIMPL : Method is not supported.

Adv_DVPAPI_GetAudioOutputMode

Syntax
HRESULT Adv_DVPAPI_GetAudioOutputMode(
LPAUDIO_OUTPUT_MODE lpMode)

Description
Retrieves the current audio mode (stereo, left, right).

Parameters
lpMode
[out] Pointer to the currently set output mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_INVALIDARG : lpMode argument is invalid.

Adv_DVPAPI_SetAudioOutputMode

Syntax
HRESULT Adv_DVPAPI_SetAudioOutputMode(
AUDIO_OUTPUT_MODE Mode)

Description
Sets the audio mode to stereo, left or right channels. When the mode is set to the left or right channel, that particular channel is heard through both speakers. See remarks section below for exceptions to this behavior.

Parameters
Mode
[in] New audio output mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
S_FALSE: Output only through chosen channel.
Remarks
Some devices do not support outputting a single channel through both speakers. On these, the audio output will come out only through the chosen channel and the method will return S_FALSE to indicate so.

Adv_DVPAPI_GetAudioOutputType

Syntax
HRESULT Adv_DVPAPI_GetAudioOutputType(
LPAUDIO_OUTPUT_TYPE lpType)

Description
Retrieves the current audio type (Dolby Surround, AC3, PCM).

Parameters
lpType
[out] Pointer to the currently set output type.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.

E_NOTIMPL: Method is not supported.
E_INVALIDARG : lpType argument is invalid.

Adv_DVPAPI_SetAudioOutputType

Syntax
HRESULT Adv_DVPAPI_SetAudioOutputType(
AUDIO_OUTPUT_TYPE Type)

Description
Sets the audio type

Parameters
Type
[in] New audio output type.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_NOTIMPL: Method is not supported.

Adv_DVPAPI_GetNavigationMode

Syntax
HRESULT Adv_DVPAPI_GetNavigationMode(
LPNAVIGATION_MODE lpMode)

Description
Retrieves the currently set navigation mode.

Parameters
lpMode
[out] Pointer to the currently set Navigation mode

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : This interface is not initialized.
E_NOTIMPL: Method is not supported.

Adv_DVPAPI_SetNavigationMode

Syntax
HRESULT Adv_DVPAPI_SetNavigationMode(
NAVIGATION_MODE Mode)

Description
Sets Navigation mode to either List (Normal) or Random.

Parameters
Mode
[in] New navigation mode

Return Value
E_FAIL: Failure.
S_OK: Success.
E_ACCESSDENIED : Mode cannot be changed on current MEDIA_FORMAT

Remarks
Only the MEDIA_FORMAT_UNKNOWN and MEDIA_FORMAT_DATA media formats can be set to NAVIGATION_MODE_LIST or NAVIGATION_MODE_RANDOM. All other media formats have predefined navigation modes that cannot be set or modified. Calling this method on these formats returns E_ACCESSDENIED.

Adv_DVPAPI_GetRepeatMode

Syntax
HRESULT Adv_DVPAPI_GetRepeatMode(
LPREPEAT_MODE lpRepeatMode)

Description
Retrieves the selected Repeat mode.

Parameters
lpRepeatMode
[out] Pointer to the currently set repeat mode.

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG : lpRepeatMode argument is invalid.

Adv_DVPAPI_SetRepeatMode

Syntax
HRESULT Adv_DVPAPI_SetRepeatMode(
REPEAT_MODE Mode)

Description
Sets the current playback repeat mode. Supported modes configure the Navigator Control to repeat the current file, playback list sequence.

Parameters
Mode
[in] New repeat mode

Return Value
E_FAIL: Failure.
S_OK: Success.
E_INVALIDARG : Mode argument is invalid.
E_ACCESSDENIED : This interface is in initialized mode.

Adv_DVPAPI_SwitchTo7420

Syntax
HRESULT Adv_DVPAPI_SwitchTo7420(
BOOL m_switch)

Description
Switch the Video output to DVP-7420 Quad output data.

Parameters
m_switch
[in] switch to DVP-7420 flag.
TRUE : switch to DVP-7420,
FALSE : switch to DVP-7410.
Default is FALSE.

Return Value
E_FAIL: Failure.
S_OK: Success.

Adv_DVPAPI_MuteOn

Syntax
HRESULT Adv_DVPAPI_MuteOn(
BOOL m_switch)

Description
Audio mute switch.

Parameters
m_switch
[in] Audio mute switch.
TRUE : Mute,
FALSE : Sound.
Default is FALSE.

Return Value
E_FAIL: Failure.
S_OK: Success.